IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:
Albornoz et al. § Filed: June 20, 2003

Serial No.: 10/600,021 § Group Art Unit: 2166

Confirmation No.: 8486 § Examiner: Khanh B. Pham

For: UNIVERSAL ANNOTATION SERVER AND INTERFACE

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office to fax number 571-273-8300 to the attention of Examiner Khanh B. Pham, or electronically transmitted via EFS-Web, on the date shown below:

February 15, 2008 /Esther Marques/
Date Esther Marques

Dear Sir:

APPEAL BRIEF

Applicants submit this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner of Group Art Unit 2166 dated September 6, 2007, finally rejecting claims 1, 5-15, 18-25, 27, 30 and 32-37. The final rejection of claims 1, 5-15, 18-25, 27, 30 and 32-37 is appealed. This Appeal Brief is believed to be timely since it is transmitted by the due date of March 3, 2008, as set by the filing of a Notice of Appeal on January 2, 2008.

Please charge the fee of $510.00 for filing this brief to:

Deposit Account No. 09-0465 / ROC920030230US1.

TABLE OF CONTENTS

Real Party in Interest

The present application has been assigned to International Business Machines Corporation, Armonk, New York.

Related Appeals and Interferences

Applicant asserts that no other appeals or interferences are known to the Applicant, the Applicant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1, 5-15, 18-25, 27, 30 and 32-37 are pending in the application. Claims 1-37 were originally presented in the application. Claims 2-4, 16, 17, 26, 28, 29 and 31 have been canceled without prejudice. Claims 1, 5-15, 18-25, 27, 30 and 32-37 stand finally rejected as discussed below. The final rejections of claims 1, 5-15, 18-25, 27, 30 and 32-37 are appealed. The pending claims are shown in the attached Claims Appendix.

Status of Amendments

All claim amendments have been entered by the Examiner, including amendments to the claims proposed after the final rejection.

Summary of Claimed Subject Matter

A. CLAIM 1 – INDEPENDENT

In the embodiments of independent claim 1, a method for exchanging information between entities on a network is provided. The method comprises identifying a plurality of annotatable data objects 121 manipulated by a plurality of applications 1201-120N on the network 127 (*See* for example *Application*, para. [0038], [0053], [0063], [0109], Figures 2, 3A-3B, 8-10C), providing a set of annotation structures 149, each associated with one or more of the annotatable data objects 121 (*See* for example *Application*, para. [0095], [0104], Figures 2, 3B) and each defining attributes of one or more user interfaces 126 for manipulating annotations for the annotatable data objects 121 (*See* for example *Application*, para. [0056], [0083], Figure 2), wherein the one or more user interfaces 126 comprise at least one graphical user interface (*See* for example *Application*, Figure 2, 5A-5H), based on an associated annotation structure 149 (*See* for example *Application*, para. [0099], Figure 3A), providing one or more transforms 750 for use in transforming annotations structures into graphical user interfaces (*See* for example *Application*, para. [0091], [0099], [0106], [0114], Figures 7A-7B, 8-10C), and providing, via an annotation management system 111 on the network 127 (*See* for example *Application*, [0036], [0040-0041], Figures 1-2), the one or more user interfaces 126 (*See* for example *Application*, [0056], Figure 2), wherein elements of each user interface 126 are dependent on the attributes defined by an associated one of the annotation structures (*See* for example *Application*, para. [0099], Figure 2), and wherein the elements are configured for user input corresponding to the manipulating of the annotations (*See* for example *Application*, para. [0056], [0083], [0099], Figure 2), wherein providing the at least one graphical user interface comprises transforming the associated annotation structure (*See* for example *Application*, para. [0091], [0099], [0106], [0114], Figures 7A-7B, 8-10C).

B. CLAIM 15 - INDEPENDENT

In the embodiments of independent claim 15, a method of creating annotations for a plurality of different type data objects manipulated by a plurality of applications is provided. The method comprises receiving a request from a user to create an annotation for a data object 502 (*See* for example *Application*, para. [0095], Figure 6A), retrieving, from a set of annotation structures 149, one or more annotation structures 149 associated with the data object (*See* for example *Application*, para. [0095], Figure 3A, 6A) and dependent, at least in part, on at least one credential of a user initiating the request (*See* for example *Application*, para. [0095]), wherein the at least one credential comprises a role of the user 504 (*See* for example *Application*, para. [0095], Figure 6A), and each annotation structure 149 containing one or more annotation fields 544 (*See* for example *Application*, para. [0076], [0080], [0084], [0087-0088], Figure 3A, 5D), generating a graphical user interface based on one of the annotation structures 506 (*See* for example *Application*, para. [0099], Figure 6A), the graphical user interface allowing entry of information corresponding to the one or more annotation fields 544 associated with the one annotation structure 149 (*See* for example *Application*, para. [0087-0088], Figure 3A, 5D), and creating an annotation record comprising the information entered, via the graphical user interface 512, for the one or more annotation fields 544 (*See* for example *Application*, para. [0100-0101], Figure 5D, 6A).

C. CLAIM 25 - INDEPENDENT

In the embodiments of independent claim 25, a computer-readable storage medium containing an executable component for managing annotations created for data objects manipulated by one or more applications on a network which, when executed by a processor, performs operations. The operations comprise receiving a request from one of the applications 120 to create an annotation for a data object 502 (*See* for example *Application*, para. [0095], Figure 2, 6A), wherein receiving a request from one of the applications 120 to create an annotation for a data object comprises receiving the request from a plug-in component 122 that provides an interface between the requesting application 120 and the executable component for managing annotations (*See* for example *Application*, para. [0040], [0046], retrieving, from a set of annotation structures 149 , one or more annotation structures 149 associated with the data object, each annotation structure 149 containing one or more annotation fields 544 (*See* for example *Application*, para. [0095], Figure 3A, 5D, 6A), generating a graphical user interface based on one of the annotation structures 506 (*See* for example *Application*, para. [0099], Figure 6A), the graphical user interface allowing entry of information corresponding to the one or more annotation fields 544 associated with the one annotation structure 149 (*See* for example *Application*, para. [0087-0088], Figure 3A, 5D), and creating an annotation record comprising the information entered, via the graphical user interface 512, for the one or more annotation fields 544 (*See* for example *Application*, para. [0100-0101], Figure 5D, 6A).

D. CLAIM 30 - INDEPENDENT

In the embodiments of independent claim 30, a system for managing annotations for different type data objects manipulated by a plurality of different type applications is provided. The system comprises an annotation database 130 for storing annotations 132 separately from the data objects 121 associated with the annotations 132 (*See* for example *Application*, para. [0037], Figures 2, 3B), a set of annotation structures 149, each defining a set of annotation fields 544 (*See* for example *Application*, para. [0076], [0080], [0084], [0087-0088], Figures 3A, 5D), an annotation server 140 configured to receive requests, issued by the applications 120, to access annotations for data objects 121 identified in the requests (*See* for example *Application*, para. [0050], [0053], Figures 3A, 3B), a set of application programming interface functions 480 providing an interface between the applications 120 and the annotation server 140 (*See* for example *Application,* para. [0045], [0065-0066], Figures 3B, 3D), and a set of application plug-ins 122, each specific to one or more of the applications 120 and configured to communicate with the annotation server 140 via the application programming interface functions 480 (*See* for example *Application*, para. [0046], [0065], Figures 3A, 3B, 3D).

Grounds of Rejection to be Reviewed on Appeal

1. Rejection of claims 1, 6-15, 18-25, 27, 30 and 32-37 under 35 U.S.C. 102(e) as being anticipated by *Gupta et al.* (U.S. Patent No. 6,956,593, hereinafter, "*Gupta*").

2. Rejection of claim 5 under 35 U.S.C. 103(a) as being unpatentable over *Gupta* as applied to claims above, and in view of *Kadel et al.* (U.S. Publication 2002/0184401, hereinafter, "*Kadel*").

ARGUMENTS

1. *Gupta* Does Not Anticipate Claims 1, 6-15, 18-25, 27, 30 and 32-37 under 35 U.S.C. 102(e).

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

Regarding claim 1, Applicants submit that *Gupta* does not teach "providing a set of annotation structures, each associated with one or more of the annotatable data objects <u>and each defining attributes of one or more user interfaces for manipulating annotations for the annotatable data objects</u>, wherein the one or more user interfaces comprise at least one graphical user interface, based on an associated annotation structure; and providing one or more transforms for use in transforming annotations structures into graphical user interfaces."

In the *Final Office Action dated September 6, 2007* (hereinafter, "*Final Office Action"*), the Examiner states that *Gupta* discloses the foregoing limitations at Col. 7 lines 27-30, Col. 9 Lines 15-25, Col. 12 Line 55 to Col. 13 Line 32 and Figs. 7-11. However, Applicants submit that the cited portions and figures are simply directed to an annotation storage structure and a user interface which enables a user to create new annotations. Specifically, Col. 7 Lines 27-30 of *Gupta* merely disclose a structure for an annotation entry, which includes a plurality of fields. The fields represent "a collection of data which define a particular characteristic of [the] annotation entry." *See Gupta,* Col. 7, Lines 37-39. Col. 9 lines 15-25 of *Gupta* simply disclose the functionality of a set Identifier field (which is one of the plurality of fields of an annotation entry). As disclosed in Col. 9, lines 15-25, the [s]et identifier(s) simply stores "data that identifies one or more sets to which [an] annotation entry belongs. Finally, Col. 12 line 55 to Col. 13 line 32 simply reference Figs. 7-11, which illustrate a user interface for creating new annotations.

Nowhere in the cited passages or figures is an annotation structure that defines attributes of a user interface disclosed. Instead, *Gupta* merely discloses a user interface to create new annotations, and the fields that comprise those annotations. Therefore, *Gupta* does not disclose "providing a set of annotation structures, each associated with one or more of the annotatable data objects and each defining attributes of one or more user interfaces for manipulating annotations for the annotatable data objects, and providing one or more transforms for use in transforming annotations structure into graphic user interface."

In response to Applicants' arguments (in particular, to the claim limitation of "providing one or more transforms for use in transforming annotations structure into graphic user interface"), the Examiner states:

> On the contrary, Gupta teaches at Fig. 3 a set of annotation structure 17, 18 in **SQL relational format** (See Col. 7 lines 5-10) are **transformed to HTML** to be rendered and displayed to the user to the browser 153.

Final Office Action, Page 9. *See also Advisory Action,* Continuation Sheet (where Examiner argues "[A]s shown at Fig. 3, the annotation structure is displayed at the client in a web brow[s]er utilizing HTML format."). However, Applicants submit that Figure 3 does not disclose a transformation of an annotation structure to HTML. Instead, Figure 3 discloses annotation stores 17, 18 that are simply accessed to compose:

> outgoing email based on annotation data, and for processing incoming email. Incoming email is received and passed to [an] ABE module 132 by [an] Email Reply Server 133. Annotation content authored at client 15, using user interface 152, is received by ABE 132 and maintained in annotation content store 17. Received meta data (control information) corresponding to the annotation content is maintained in annotation meta data store 18. The annotation content and meta data can be stored in any of a variety of conventional manners, such as in SQL relational databases[.]

See Gupta, Col 6., Lines 65 through Col 7., Lines 1-10.

Accordingly, nowhere in the cited section or any other section of *Gupta*, is providing one or more transforms for use in transforming annotations structure into graphic user interface disclosed.

Therefore, Applicants respectfully request the rejection of independent claim 1, and the claims that depend therefrom, be reversed and the claims be allowed.

Regarding claim 15, Applicants submit that *Gupta* does not teach “retrieving one or more annotation structures associated with a data object and dependent, at least in part, on at least one credential of a user initiating the request, wherein the at least one credential comprises a role of the user” and “generating a graphical user interface based on one of the annotation structures.”

In the *Final Office Action*, the Examiner argues that *Gupta* teaches retrieving one or more annotation structures associated with the data object and dependent on a role of the user, and generating a graphical user interface based on one of the annotation structures at Col. 12 line 55 to Col. 33 line 32. However, the cited portion is simply directed to an “Add New Annotation” dialog box, where the user can select an annotation set identifier from either a predefined set or a new set created by the user. *Gupta*, Col. 12 lines 60-66.

In contrast, the present claims disclose retrieving one or more annotation structures associated with a data object and <u>generating a graphical user interface based on one of the annotation structures</u>. In other words, the Examiner’s rejection would require that the “Add new annotation” dialog box of *Gupta* be generated based on the retrieved annotation structure. Certainly nowhere in the cited passages or any other portion of *Gupta* is retrieving one or more annotation structures associated with a data object and generating a graphical user interface based on one of the annotation structures.

Therefore, Applicants respectfully request the rejection of independent claim 15, and the claims that depend therefrom, be reversed and the claims be allowed.

Claim 25 has similar limitations. Therefore, by the same reasons stated above, Applicants respectfully request the rejection of independent claim 25, and the claims that depend therefrom, be reversed and the claims be allowed.

Regarding claim 30, Applicants submit that *Gupta* does not teach "a set of application plug-ins, each specific to one or more of the applications and configured to communicate with the annotation server via the application programming interface functions."

In the *Final Office Action*, the Examiner makes a conclusory statement that this claim contains similar limitations to other claims that were rejected based on *Gupta*. In his response to Applicants' previous arguments, the Examiner states that:

> as seen in Fig. 3, Gupta teaches the MAWS 130, which is a plug-in for the IIS module 135, that provides an interface between requesting application(i.e., Web Browser 153) and the executable component for managing annotation(i.e., ABE 132). Further, the user interface 152 could also be considered "a plug-in component" for the web browser 153, which sends a request to annotation server 10 and provides an interface between the requesting application (i.e., browser 153) and the executable component for managing annotation (i.e., "annotation server 10).

Final Office Action, Page 10. However, as illustrated in Figure 3 of *Gupta,* the MAWS 130 is part of the annotation server 10 (*See also* Col. 6 lines 52-63), whereas claim 30 discloses that a set of application plug-ins are configured to <u>communicate with the annotation server via the application programming interface functions (API)</u>. Thus, the application plug-ins are not part of the annotation server (as clearly illustrated in Fig. 2 of the Application).

The Examiner further argues that the user interface 152 could be considered as "a plug-in component." Applicants submit that the application plug-in is configured to communicate with the annotation server via an API. In contrast, the user interface 152 in Fig. 3 of *Gupta* communicates with "an annotation back end (ABE) module 151, which translates user actions into commands destined for [a] server 10, and an HTTP services module 150, which manages communication destined for the server 10. *Gupta*, Col. 6 lines 29-33. Clearly, the user interface 152 does not communicate with the annotation server 10 via an API, but through an ABE module and an HTTP services module.

Additionally, the Examiner fails to identify a single relevant portion of *Gupta* that discloses that each application plug-in is specific to one or more of the applications as disclosed in the present claim. Therefore, the cited section of *Gupta* does not disclose the application plug-in component as recited in the present claim. Thus, *Gupta* does not teach "a set of application plug-ins, each specific to one or more of the applications and configured to communicate with the annotation server via the application programming interface functions."

Therefore, Applicants respectfully request the rejection of independent claim 30, and the claims that depend therefrom, be reversed and the claims be allowed.

2. The combination of *Gupta* in view of *Kadel* Does Not Render Claim 5 obvious.

Claim 5 stands rejected under 35 U.S.C. 103(a) as being unpatentable over *Gupta* in view of *Kadel*.

Applicants respectfully submit that the rejection has been overcome as described above. Accordingly, Applicants respectfully request the rejection be reversed and the claim be allowed.

CONCLUSION

The Examiner errs in finding that:

1. Claims 1, 6-15, 18-25, 27, 30 and 32-37 are anticipated by *Gupta*; and

2. Claim 5 is unpatentable over *Gupta* in view of *Kadel.*

Withdrawal of the rejections and allowance of all claims is respectfully requested.

Respectfully submitted, and
S-signed pursuant to 37 CFR 1.4,

/Gero G. McClellan, Reg. No. 44,227/

Gero G. McClellan
Registration No. 44,227
Patterson & Sheridan, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Appellants

CLAIMS APPENDIX

1. (Previously Presented) A method for exchanging information between entities on a network comprising:

identifying a plurality of annotatable data objects manipulated by a plurality of applications on the network;

providing a set of annotation structures, each associated with one or more of the annotatable data objects and each defining attributes of one or more user interfaces for manipulating annotations for the annotatable data objects, wherein the one or more user interfaces comprise at least one graphical user interface, based on an associated annotation structure;

providing one or more transforms for use in transforming annotations structures into graphical user interfaces; and

providing, via an annotation management system on the network, the one or more user interfaces, wherein elements of each user interface are dependent on the attributes defined by an associated one of the annotation structures and wherein the elements are configured for user input corresponding to the manipulating of the annotations, wherein providing the at least one graphical user interface comprises transforming the associated annotation structure.

2-4. (Cancelled)

5. (Previously Presented) The method of claim 1, wherein the one or more transforms comprise one or more Extensible Stylesheet Language transforms.

6. (Original) The method of claim 1, further comprising installing one or more plug-in components for interfacing between the one or more applications and the annotation management system.

7. (Original) The method of claim 6, further comprising installing an annotation broker on the one or more client computers, the annotation broker providing an interface between one or more of the plug-in components and the annotation server.

8. (Original) The method of claim 1, further comprising installing a set of application programming interface functions for the annotation management system, callable from the one or more applications.

9. (Original) The method of claim 8, wherein the set of application programming interface functions comprise functions for manipulating annotations.

10. (Original) The method of claim 8, wherein the set of application programming interface functions comprise functions for retrieving annotations for a specified data object.

11. (Original) The method of claim 8, wherein the set of application programming interface functions include functions for retrieving an indication of data objects described by an annotation.

12. (Original) The method of claim 8, wherein the set of application programming interface functions comprise at least one function for retrieving an indication of the plurality of annotatable data objects.

13. (Original) The method of claim 1, wherein providing the annotation structures comprises selecting, for each annotation structure, one or more annotation fields to include in the annotation structure.

14. (Previously Presented) The method of claim 13, wherein at least some of the one or more user interfaces include elements allowing a user to enter information corresponding to one or more annotation fields included in an associated annotation structure.

15. (Previously Presented) A method of creating annotations for a plurality of different type data objects manipulated by a plurality of applications, comprising:

receiving a request from a user to create an annotation for a data object;

retrieving, from a set of annotation structures, one or more annotation structures associated with the data object and dependent, at least in part, on at least one credential of a user initiating the request, wherein the at least one credential comprises a role of the user, and each annotation structure containing one or more annotation fields;

generating a graphical user interface based on one of the annotation structures, the graphical user interface allowing entry of information corresponding to the one or more annotation fields associated with the one annotation structure; and

creating an annotation record comprising the information entered, via the graphical user interface, for the one or more annotation fields.

16-17. (Cancelled)

18. (Original) The method of claim 15, wherein a plurality of annotation structures are associated with the data object and the method further comprises:

presenting, to a user, the plurality of annotation structures associated with the data object;

receiving, from the user, a selection of one of the plurality of annotation structures; and

generating the graphical user interface based on the selected annotation structure.

19. (Original) The method of claim 18, further comprising receiving, from the user, a selected role in which the user has chosen to act.

20. (Original) The method of claim 19, wherein the plurality of annotation structures presented to the user is dependent on the selected role.

21. (Original) The method of claim 19, further comprising:

retrieving, via an application programming interface, a plurality of roles associated with the user; and

presenting, to the user, the plurality of roles associated with the user.

22. (Original) The method of claim 15, wherein retrieving one or more annotation structures associated with the data object comprises passing an application programming interface function at least an indication of the data object.

23. (Original) The method of claim 22, wherein retrieving the one or more annotation structures associated with the data object further comprises passing the application programming interface function at least one credential of a user.

24. (Original) The method of claim 22, wherein the at least one user credential comprises at least one of a role and a user identification.

25. (Previously Presented) A computer-readable storage medium containing an executable component for managing annotations created for data objects manipulated by one or more applications on a network which, when executed by a processor, performs operations comprising:

receiving a request from one of the applications to create an annotation for a data object, wherein receiving a request from one of the applications to create an annotation for a data object comprises receiving the request from a plug-in component that provides an interface between the requesting application and the executable component for managing annotations;

retrieving, from a set of annotation structures, one or more annotation structures associated with the data object, each annotation structure containing one or more annotation fields;

generating a graphical user interface based on one of the annotation structures, the graphical user interface allowing entry of information corresponding to the one or more annotation fields associated with the one annotation structure; and

creating an annotation record comprising the information entered, via the graphical user interface, for the one or more annotation fields.

26. (Cancelled)

27. (Previously Presented) The computer-readable medium of claim 25, wherein receiving a request from one of the applications to create an annotation for a data object comprises receiving the request from an annotation broker that provides an interface between plug-in components of one or more applications and the executable component for managing annotations.

28-29. (Cancelled)

30. (Previously Presented) A system for managing annotations for different type data objects manipulated by a plurality of different type applications, comprising:

an annotation database for storing annotations separately from the data objects associated with the annotations;

a set of annotation structures, each defining a set of annotation fields;

an annotation server configured to receive requests, issued by the applications, to access annotations for data objects identified in the requests;

a set of application programming interface functions providing an interface between the applications and the annotation server; and

a set of application plug-ins, each specific to one or more of the applications and configured to communicate with the annotation server via the application programming interface functions.

31. (Cancelled)

32. (Original) The system of claim 30, wherein the annotation server is configured to retrieve, via one or more application programming function calls, annotations associated with a data object identified in a request.

33. (Previously Presented) The system of claim 30, wherein the annotation server is configured to:

retrieve, via a first application programming function call, one or more annotation identifications associated with the data object identified in the request; and

using the annotation identifications, retrieve, via a second application programming interface function call, the corresponding annotations from the annotation store.

34. (Original) The system of claim 30, wherein the annotation server is configured to:

retrieve, via an application programming interface function call, a list of one or more roles associated with a user; and present, to the user, the one or more roles associated with the user.

35. (Original) The system of claim 33, wherein the annotation server is further configured to:

receive, from the user, a selected one of the one or more roles associated with the user; and

indicate to the system, via an application programming interface function call, the role selected by the user.

36. (Previously Presented) The system of claim 30, wherein the annotation server is configured to:

retrieve, via an application programming interface function call, annotation structures associated with data objects identified in requests; and

transform the annotation structures into graphical user interfaces for creating annotations for the data objects.

37. (Original) The system of claim 36, wherein the annotation server is further configured to retrieve, via an application programming interface function call, one or more transforms associated with an annotation structure for use in transforming the annotation structure into a graphical user interface.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.